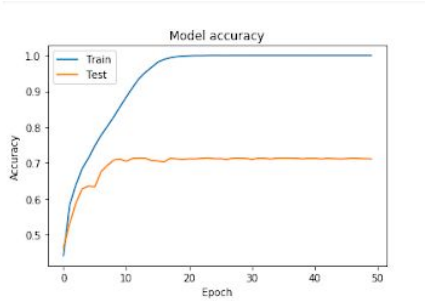
Machine Learning Assignment 2

Mayank Singh Kushwaha
2017CS50413

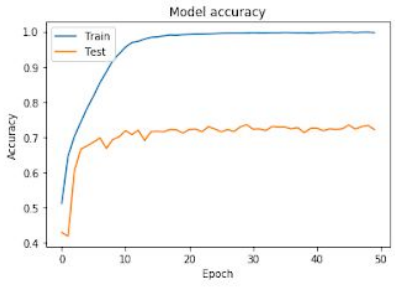
Part A:-

Choosing Optimizer:

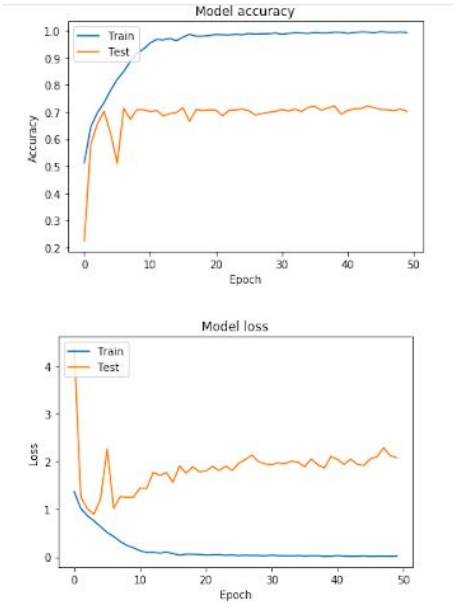
a).



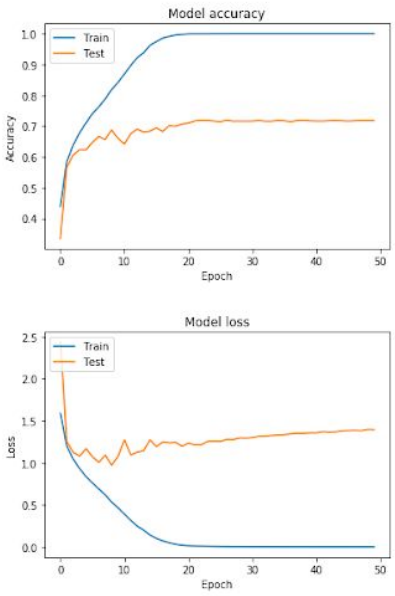
b)

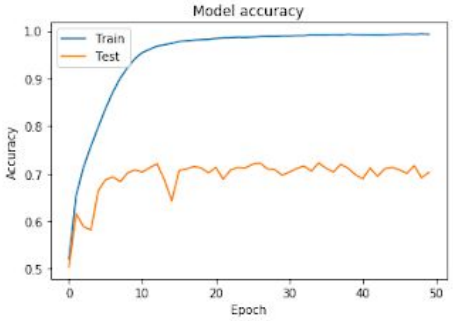


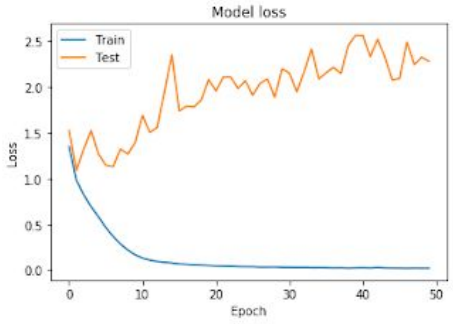
c)



d)







e)

a) AdaDelta b) AdaGrad c) Adam d) SGD e) RMS

I first chose the optimizer function I plotted the graph of validation accuracy, loss and training accuracy, loss wrt number of epochs.
From the graph above, I decided to use the AdamGrad optimizer. Reason for choosing the optimizer was it was giving high accuracy among the others although it overfit if we run for more number of epochs, we will choose the epochs according to the graph.

Epochs

Number of epochs chosen is 17 because after that we can see from the graph the data starts to overfit.

Batch Size:

I submitted my prediction file on moodle and selected the batch that gave the highest accuracy.

1) 32 - 73.28
2) 64 - 72.65

3) 128 - 72.66

And, clearly best accuracy came for batch size = 32.

*All the testing for choosing the best parameters was done on kaggle and best architecture was run on hpc.

Reason For Increase in Loss Function value and increase in accuracy

When the predictions get more confident, loss gets better even though accuracy stays the same. The model is thus more robust, as there's a wider margin between classes.

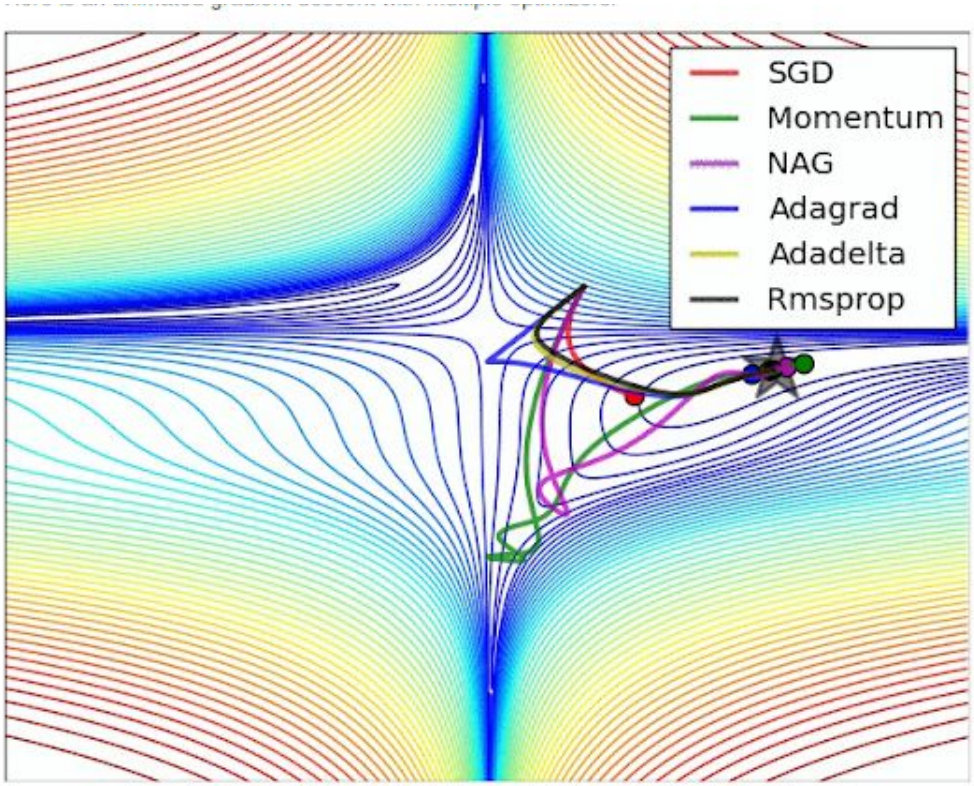
If the model becomes over-confident in its predictions, a single false prediction will increase the loss unproportionally compared to the (minor) drop in accuracy. An over-confident model can have good accuracy but bad loss. I'd assume over-confidence equals over-fitting.

Imbalanced distributions: if 90% of the samples are "apples", then the model would have good accuracy score if it simply predicts "apple" every time.

PartB:-

From the previous part I knew I had to use Adagrad optimizer, batch size = 32. I experimented with few architecture. The things I focused mainly was to implement dropout, momentum, Feature Engineering, Activation Function.

AdaGrad:



Feature Engineering:

I decided to include corresponding Grayscale image in my dataset.
Convert to grayscale image Y = 0.2989 R + 0.5870 G + 0.1140 B

Why I chose “elu” activation function over “Relu”?

- ELU becomes smooth slowly until its output equal to -α whereas RELU sharply smoothes.

- Unlike to ReLU, ELU can produce negative outputs.

Why I used Momentum?

As Adagrad becomes very slow at the end when it approaches the minima then momentum help us to get closer to the minima. And due to this reason I have to use momentum to increase my prediction.

Why use Dropout?

Dropout is used so that deep model does not overfit the training example easily. As my final architecture is deep hence I had to implement Dropout to improve my accuracy on the test data.

Architecture:

I have to insert more Convolution layers as the previous architecture was giving around 79.82% accuracy but after including one more layer the accuracy improved to 83.67.

Best Architecture:-

CONV1: 32 output, input image with 4 channels, activation function(elu), kernel size is (3, 3)
CONV2: 32 output, activation function(elu), kernel size is (3, 3), Dropout(0.2)
POOL: 2x2 Max Pooling layer.
CONV3:
CONV4: 64 output, activation function(elu), kernel size is (3, 3), Dropout(0.3)
POOL: 2x2 Max Pooling layer.
CONV5:
CONV6:128 output, activation function(elu), kernel size is (3, 3), Dropout(0.4)
POOL: Max Pooling layer.
FC1: Fully connected layer with 512 outputs.

FC2: Fully connected layer with 256 outputs.
SM: Softmax layer for classification

The parameters value selected by testing them on kaggle.